

What is claimed is:

- 1 1. A method for modeling a web server, comprising:
2 identifying a plurality of sub-systems for the server;
3 representing each sub-system as a queue, with each queue operably coupled
4 together; and
5 iteratively adjusting an arrival rate and a service time for each queue to
6 account for performance by other queues.
- 1 2. The method of claim 1, wherein said plurality of sub-systems comprises one
2 or more of a set comprising a transaction control protocol/internet protocol sub-
3 system, a hypertext transfer protocol sub-system, an input/output sub-system, and an
4 active script component sub-system.
- 1 3. The method of claim 1, wherein each sub-system is modeled as a finite-
2 buffer, finite server queueing system.
- 1 4. The method of claim 2, wherein said transaction control protocol/internet
2 protocol sub-system comprises a first finite listen queue served by a listener daemon.

1 5. The method of claim 2, wherein said hypertext transfer protocol sub-system
2 comprises a second finite listen queue served by one or more multi-threaded
3 hypertext transfer protocol daemons with N_{http} separate server threads.

1 6. The method of claim 2, wherein said input/output sub-system comprises a
2 finite number N_{buf} of network buffers served by an input/output controller.

1 7. The method of claim 6, wherein said input/output controller serves each
2 network buffer using a polling system.

1 8. The method of claim 2, wherein said transaction control protocol/internet
2 protocol sub-system TCP/IP is represented as an $M(\lambda_{\text{file}}) / M(\tau_{\text{tcp}}) / N_{\text{tcp}} / 0$ blocking
3 system.

1 9. The method of claim 2, wherein said hypertext transfer protocol sub-system
2 is represented as an $M(\lambda_{\text{http}}) / M(\tau_{\text{http}}) / N_{\text{http}} / Q_{\text{http}}$ queueing system.

1 10. The method of claim 2, wherein said input/output sub-system is represented
2 as an $M(\lambda_{\text{buf}}) / M(\tau_{\text{buf}}) / N_{\text{buf}} / \infty$ queueing system.

- 1 11. A method for modeling a web server, comprising:
- 2 (a) identifying for the server a transaction control protocol/internet
- 3 protocol (TCP/IP) sub-system, a hypertext transfer protocol (HTTP) sub-system, and
- 4 an input/output (I/O) sub-system;
- 5 (b) representing each sub-system as a queuing system;
- 6 (c) computing an upper bound performance for said I/O sub-system by
- 7 assuming a first predetermined blocking value for said TCP/IP sub-system and
- 8 HTTP sub-system;
- 9 (d) computing an upper bound performance for said TCP/IP sub-system
- 10 and HTTP sub-system by assuming a first predetermined I/O sub-system waiting
- 11 time;
- 12 (e) computing a lower bound I/O performance by assuming a second
- 13 predetermined blocking value for said TCP/IP sub-system and HTTP sub-system;
- 14 (f) computing a lower bound performance for said TCP/IP sub-system
- 15 and HTTP sub-system by assuming a second predetermined I/O sub-system waiting
- 16 time; and
- 17 (g) repeating steps (c) - (f) to generate successively tighter bounds until
- 18 convergence.

1 12. A machine-readable medium whose contents cause a computer system to
2 model a web server, by performing the steps of:
3 identifying a plurality of sub-systems for the server;
4 representing each sub-system as a queue, with each queue operably coupled
5 together; and
6 iteratively adjusting an arrival rate and a service time for each queue to
7 account for performance by other queues.

1 13. The machine-readable medium of claim 12, wherein said plurality of sub-
2 systems comprises one or more of a set comprising a transaction control
3 protocol/internet protocol sub-system, a hypertext transfer protocol sub-system, an
4 input/output sub-system, and an active script component sub-system.

1 14. The machine-readable medium of claim 12, wherein each sub-system is
2 modeled as a finite-buffer, finite server queueing system.

1 15. The machine-readable medium of claim 13, wherein said transaction control
2 protocol/internet protocol sub-system comprises a first finite listen queue served by a
3 listener daemon.

1 16. The machine-readable medium of claim 13, wherein said hypertext transfer
2 protocol sub-system comprises a second finite listen queue served by one or more
3 multi-threaded hypertext transfer protocol daemons with N_{http} separate server
4 threads.

1 17. The machine-readable medium of claim 13, wherein said input/output sub-

1 system comprises a finite number N_{buf} of network buffers served by an input/output
2 controller.

1 18. The machine-readable medium of claim 17, wherein said input/output
2 controller serves each network buffer using a polling system.

1 19. The machine-readable medium of claim 13, wherein said transaction control
2 protocol/internet protocol sub-system TCP/IP is represented as an $M(\lambda_{file}) / M(\tau_{tcp}) /$
3 $N_{tcp} / 0$ blocking system.

1 20. The machine-readable medium of claim 13, wherein said hypertext transfer
2 protocol sub-system is represented as an $M(\lambda_{http}) / M(\tau_{http}) / N_{http} / Q_{http}$ queueing
3 system.

1 21. The machine-readable medium of claim 13, wherein said input/output sub-
2 system is represented as an $M(\lambda_{buf}) / M(\tau_{buf}) / N_{buf} / \infty$ queueing system.

- 1 22. A machine-readable medium for modeling a web server, comprising:
- 2 (a) identifying for the server a transaction control protocol/internet
- 3 protocol (TCP/IP) sub-system, a hypertext transfer protocol (HTTP) sub-system, and
- 4 an input/output (I/O) sub-system;
- 5 (b) representing each sub-system as a queuing system;
- 6 (c) computing an upper bound performance for said I/O sub-system by
- 7 assuming a first predetermined blocking value for said TCP/IP sub-system and
- 8 HTTP sub-system;
- 9 (d) computing an upper bound performance for said TCP/IP sub-system
- 10 and HTTP sub-system by assuming a first predetermined I/O sub-system waiting
- 11 time;
- 12 (e) computing a lower bound I/O performance by assuming a second
- 13 predetermined blocking value for said TCP/IP sub-system and HTTP sub-system;
- 14 (f) computing a lower bound performance for said TCP/IP sub-system
- 15 and HTTP sub-system by assuming a second predetermined I/O sub-system waiting
- 16 time; and
- 17 (g) repeating steps (c) - (f) to generate successively tighter bounds until
- 18 convergence.